



CHAPTER 3

➤ BY :

➤ Vishu Goyal

➤ Etisha Jain

LINUX 3.1 Essential File Management Tasks

- `ls` to list all the files and directory
- `mkdir` allows you to create a directory
- `mkdir new/scripts` now allowed to create like recursively
- `mkdir -p new/scripts` are allowed to create a directory.
- `cp` to copy
- `cp --help | less`
- `cp /etc/hosts .`
- `cp /etc/h* .`
- `cp -r /etc/h*` it will copy all the directory and folders recursively
- `mv` to move
- `mv script* scripts/` / represents the folder
- `mv script* scripts` without slash it is represent a file.
- `rmdir` it remove the empty directory
- `rm -rf <folder_name>` it removes the directory with recursively and forcefully

using `rm` is very danger (it will remove all the entire file system)

- `rm -rf /`
- `rm --no-preseve-root`

LINUX 3.2 Finding Files

- **ls** it is used to list files, not to find files
- **which** looks for binaries in **\$PATH**
- **locate** uses a database, built by **updatedb** to find files in a database. If you just create a file and not update a database then it won't show that through the locate cmd because it was not updated yet.
- **find** it is the most flexible tool that allows you to find files based on many criteria.

Practical :

which work depend on \$PATH

- which useradd
- echo \$PATH it will show all the path variable where which cmd searched the binary files
- let's just do one thing as a practical
- create a file
- vim hello
- #!/bin/bash
- echo hello world
- chmod +x hello
- type hello command not found error
- ./hello it will execute that script file in the current folder.
- output -> hello world
- which hello
- output -> hello is not in the \$PATH directory.

which work depend on database

- locate useradd it error got then your database is not updated
- updatedb it will create a database or update.
- locate useradd
- now you will create a file and not update a database
- then you will not find anything because locate works on the database
- we want to update a database to locate the file
- updatedb command will run to update the database.
- now locate the file
- locate <file_name>
- make sure you will run a updatedb command once a day to update your database

Practical:

Find based on many criteria and it is best to find anything but it is also help to use with different combinations of option and commands.

- find / -name "hosts"
- find / -type f
- find / -size +100M
- find / -type f -size +100M
- find / -user student
- find / -user student | grep wc
- find /etc -exec grep -l student {} \;
- find /etc -exec grep -l student {} \; 2>/dev/null
- grep -l student /etc/* 2>/dev/null
- find /etc/ -size +100c -exec grep -l student {} \;
- find /etc/ -size +100c -exec grep -l student {} \; -exec cp {} /tmp \; 2>/dev/null

In the last command, first find all the files of +100c then it will execute the other command. It will copy the output to the \tmp file and if the error will come then it goes to the nothing.



LINUX 3.3 Understanding the Mounts System

- To access a device, it must be connected a directory
- This is known as mounting the device.
- The linux filesystem typically uses multiple mounts.
- Different types of data typically are on different devices for multiple reasons
- this is all done becoz of :
 - security
 - manageability
 - specific mount options



3.4 Understanding Links

- Link is the most fundamental concept in linux
- Links are the pointers to files in a different location
- by using the links you will make your file system little bit easier.
- Compare links to the shortcuts.
- Links can be useful to make the same file available on multiple locations
- links uses hard links and symbolic links
- you can create a hardlink through `ln` command
- create a symbolic link by using `ln -s` command

Files are the collections of Blocks which is in hard disk
every files linux has its own inode and every inode has its own number
and inodes are kept in the inode table and every file system has its own inode table
the inode number is not convinient for the normal user thats why we will give a name.

we can use multiple name for a single inode that is hard links and inode generate a inode counter
and if one name is removed then nothing is wrong the inode counter is decresed.

on top of this hard link we can create a symbolic link
this link is more flexible becex symbolic link is used for cross device and directory

one problem is done if some hardlink is destroy then the symbolic link is pointed to that but it
doesn't points more. then the symbolic link is invalid.


LINUX 3.5 Working With Links

- `ls -il /etc/host` option `-l` is for long and `-i` is to display the inode number.
- the inode number is unique.
- `ln /etc/hosts /root/hardhosts` it will create a hard link
- `ls -il /etc/hosts /root/hardhosts`
- The upper cmd will show both file have same size becoz the pointing to the device.
- Now will create a symbolic links,
- `ln -s /etc/hosts symhosts`
- `ls -il /etc/hosts /root/hardhosts /root/symhosts`
- see carefully symbolic links have size less then its original files.
- it will shows in blue color
- now want to remove a `/etc/hosts` file
- `rm -f /etc/hosts`
- `ls -il /etc/hosts /root/hardhosts /root/symhosts`
- it will show like a blinking becoz `/etc/hardhost` does not exit and red color
- `cat symhosts`
- move to the root directory
- `cd /`
- `ls -l` it will shows some linked files

LINUX 3.6 Working with tar

- tar is the Tap Archiver and was created a long time ago
- by default, it doesn't compress data
- at the time of tar was created, it was just streaming data to a backup tape
- no compression was involved
- so the idea was just to put the content of all files behind one another on the backup tape.
- Basic use of tar is to compress, extract or list

- **tar -cvf my_archive.tar /home/etc**
- **-c** stands for create
- **-v** stands for verbose
- **-f** stands for file
- then name of the files.tar
- then the directory or files you want to create a tar
- **tar -tvf** will show content of an archive and it is just for verification purposes.
- **-t** will show the content of the archive file
- **tar -xvf my_archive** extracts to the current directory.
- Use **-C** to switch the output path
- to add compression, use
- **-z** for gzip
- **-j** for bzip2
- **-J** for xz utility

- 
- Eg: **tar -cvf ~/mytar.tar /home/etc**
 - to check the content of the archive file
 - **tar tvf mytar.tar | less**
 - **tar xvf mytar.tar** to extract the files and also keep the original archive
 - **tar xvf mytar.tar -C /tmp** all the extracted files move inside the /tmp folder.

Note: In linux we are not give the extension of the files explicitly becoz in linux it works on magic code.

Magic code will be written of every file.

- **mv mytar.tar mytar** to rename the file
- **file mytar** to check which type of file is it..



LINUX 3.7 Understanding Compression Utilities

A wide range of compression solution is available for linux

- **bzip** is still the most common compression utility.
 - **bzip2** is an alternative utility.
 - **zip** is also available and has windows compatible syntax, its difficult to use it.
 - **xz** is showing up more often as well.
-
- First mytar.tar files is about 37 MiB
 - `gzip mytar` is about 7.2 MiB original files is deleted
 - `gunzip mytar.gz`
 - `gzip -k mytar` is about 7.2 MiB original files is keep becoz of option -k
 - `bzip2 mytar` is about 5.4 MiB original files is deleted
 - `bzip2 -k mytar` is about 5.4 MiB original files is keep becoz of option -k
 - `xz -k mytar` is about 4.3 MiB original files is keep becoz of option -k



THANK



YOU